REMARKS

In the Official Action mailed on **17 July 2006**, the Examiner reviewed claims 1-25. Claims 1, 6-11, 13, and 18-23 were rejected under 35 U.S.C. §102(b) as being anticipated by "Speculative Lock Elision: Enabling Highly Concurrent Multithreaded Execution" by Ravi Rajwar and James R. Goodman (hereinafter "Rajwar"). Claim 12 and 24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Rajwar in view of common art. Claims 1, 6-11, 13, 18-23 and 25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Rajwar in view of "Enhancing Software Reliability with Speculative Threads" by Jeffrey Oplinger and Monica S. Lam (hereinafter "Lam").

Rejections under 35 U.S.C. § 102(b) and 35 U.S.C. § 103(a)

Independent claims 1 and 13 were rejected as being anticipated by Rajwar. Applicant respectfully points out that Rajwar discloses a hardware-based mechanism for dynamically predicting unnecessary lock operations and eliding such operations (see Rajwar, page 295, col. 1, lines 29-35). This is implemented "entirely in the microarchitecture, without instruction set support and without system-level modifications ... and is transparent to programmers" (see Rajwar, page 295, col. 2, lines 16-19).

In contrast, the present invention provides a special "start transactional execution" (STE) instruction, which can be explicitly executed before a process executes a critical code section. The instant application discloses that the processor instruction set is augmented to include the STE instruction (see paragraph [0059] of the instant application). This **STE instruction performs a series of operations before the process starts executing the critical code section, and these operations specify possible actions that can be taken if the critical section code fails** (see paragraph [0081]-[0083] and FIGS. 3 and 4 of the instant application).

Examiner has also added a 35 U.S.C. §103(a) rejection under Rajwar in view of Lam. Applicant respectfully notes that Lam discloses programming constructs that assist in the execution of speculative threads. However, there is nothing within Lam, explicit or implicit, that discloses a transactional programming construct that specifies the action of acquiring a lock on the block of instructions as a possible execution path if the transactional execution of the block of instructions following the construct fails.

In contrast, the present invention teaches that the system attempts to transactionally re-execute the critical section, and if re-execution attempts are unsuccessful, the system **can revert back to the conventional technique of acquiring a lock** (see paragraphs [0064] and [0083] of the instant application).

Accordingly, Applicant has amended independent claims 1, 13, and 25 to clarify that the present invention provides a STE instruction that specifies an action to take if transactional execution of a critical section of code following the instruction fails, and also clarifies that one such action can be acquiring a lock on the block of instructions. These amendments find support in FIGs. 3 and 4, and in paragraphs [0064] to [0081]-[0083] of the instant application. Applicant has also canceled claims 2, 4, 14, and 16 in accordance with the above changes. Dependent claims 3, 5, 15, and 17 have been amended to correct antecedent basis.

Hence, Applicant respectfully submits that independent claims 1, 13, and 25 as presently amended are in condition for allowance. Applicant also submits that claims 3, 5-12, which depend upon claim 1, and claims 15, 17-24, which depend upon claim 13, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

CONCLUSION

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By

Edward J. Grundler
Registration No. 47,615

Date: 31 July 2006

Edward J. Grundler
PARK, VAUGHAN & FLEMING LLP
2820 Fifth Street
Davis, CA 95618-7759
Tel: (530) 759-1663
FAX: (530) 759-1665
Email: edward@parklegal.com